VMware

Alberto Farronato, Director of Product Marketing, Storage
Christos Karamanolis - Chief Architect and Principal Engineer, Storage and Availability

VMware vision for SDS
Alberto

Challenges today- not commodity, low utilisation, over provisioning
Specialised, expensive hw - static class of service, rigid provisioning, lack of control
device-centric silos
complex processes

From hw-centric to app-centric - policy, driven automation, common across arrays, dynamic control
new data plane - from specialise to industry std - server SAN, Flash accelerated, distributed

Leverage the hypervisor
VMware storage policy-based mgmt
- app-centric storage automation
- common mgmt across heterogeneous arrays

VSAN
- hyper-converged arch
- data persistence delivered from the hypervisor

Announced Q1
VMware VSAN 6.0
- all-flash arch
- went data services
- broader hw support

Virtual SAN technical overview
Christos Karamanolis (@XtosK)

product goals
1. customer - vsphere admin
2. compelling TCO
- CAPEX - capacity, ease of mgmt
- OPEX - ease of mgmt
3. SDS for VMware
- strong integration with all VMware products
- same operational experience and tools

Architectural Principles
1. Hyper-converged
- compute + storage scalability
- unobtrusive to existing DC arch

- distributed sw running on every host
- pool local storage (flash + HDD) on hosts (virtual shared datastore)
- symmetric arch - no SPoF, no bottleneck

The hypervisor opens up new opportunities
the virtualisation platform:
- visibility to individual VMs and app storage
- Manages all apps resource reqs
- sits directly in the IO path
- global view of underlying infrastructure
- supporting extensive HCL

2. Critical paths in ESX kernel
Cluster Service
- fast failure detection
- high perf (esp writes)
Data Path
- low latency
- minimal CPU per IO
- Minimal Mem consumption
- physical access to devices
= minimal impact on consolidation rates

3. Optimized internet protocol
ESXi: both “consumer” and “producer” of data - no need for std data access protocol
Per-object coordinator = client
- distributed “metadata server”
- transactions span only object distribution
Efficient reliable data transport (RDT)
- protocol agnostic (now TCP/IP)
- RDMA friendly
Standard protocol for external access?

4. Two tiers of storage- Hybrid
Optimise cost of physical storage resources
- HDDS: cheap capacity, expensive IOPS
- Flash: expensive capacity, cheap IOPS
Combine best of both worlds
- performance from flash (read cache + write back)
- capacity from HDD (capacity tier)
Optimise workload per tier
- random IO to flash (high IOPS)
- sequential IO to HDD (high throughput)

storage organised in disk groups (flash device and magnetic disks) - up to 5 disk groups, 1 SSD + 7 HDDs - this is the fault domain
70% of flash is read cache, 30% is write buffer
Writes are accumulated, then staged in a magnetic disk-friendly fashion
proximal IO - writing blocks within a certain number of cylinders. filesystem on the magnetic disks is slightly different to the one on the SSDs
Use the back-end of the virsto filesystem, don’t use the log-structure filesystem

component

5. Distributed caching
Flash device: cache of disk group
- 70% read cache, 30% write-back buffer
No caching on “local” flash where VM runs
- flash latencies 100x network latencies
- no data transfers, no perf hit during VM migration
- better overall flash utilisation (most expensive resource)
Use local cache when it matters
- in-memory CBRC (RAM << Network latency)
- lots of block sharing (VDI)
- More options in the future …

deduplicated RAM-based caching

6. object-based storage
VM consists of a number of objects
- each object individually distributed
A storage platform designed for

[gap]

VSAN doesn’t know about VMs and VMDKs
Up to 62TB useable
Single namespace, multiple mount points
VMFS created in sub-namespace
The VM Home directory object is formatted with VMFS to allow a VM’s config files to be stored on it. Mounted under the root dir vsanDatastore
- availability policy reflected on number of replicas
- perf policy may include a stripe width per replica
- object “components” may reside in different disks and / or hosts

7. VSAN cluster = vSphere cluster
Ease of management
- piggyback on vSphere management workflow, e.g. EMM
- ensure coherent configuration of hosts in vSphere cluster
Adapt to customer DC architecture
- network topology constraints

Maintenance mode - planned downtime
3 options:
- ensure accessibility
- full data migration
- no data migration

HA Integration

8. VM-centric montiroing and troubleshooting
VMODL APIs
- configure, manage, monitor

Policy compliance reporting
combination of tools for monitoring in 5.5
- CLI commmands
- Ruby vSphere console
- VSAN observer
More to come soon

9. Real *software* defined storage
Software + hardware - component based (individual components), virtual SAN ready node (40 OEM validated server configs ready for VSAN deployment)
VMware EVO:RAIL - Hyper-converged infrastructure

It's a big task to get all of this working with everything (supporting the entire vSphere HCL)

VMware Virtual SAN Compatibility Guide (VCG)

Maximum Flexibility (components) -> Maximum ease of use (EVO)

All flash configs in VSAN 6 will require an additional license

What's new in VSAN 6
All flash architecture
4x perf, 2x scale
enterprise data services
broader hardware support

VSAN 6: All-flash VSAN: 2 tiers
optimise cost of flash
- high endurance, expensive capacity
- low-endurance, cheap capacity
Most cost-efficient model
- small expensive high-endurance write cache
- large inexpensive low-endurance capacity tier
Optimise workload per tier
- keep frequently written data in write cache
- lots of read IOPS from capacity tier

May revisit in the future …

VSAN 6: Efficient "vsanSparse" snapshots and clones
New redirect-on-write snapshot
utilizing "VsanFS" (verso) sparse format
- same on-disk format for delta disks
- in-memory cache of written extents
- GWE of hierarchy upon cache miss
Minimal performance degradation
greater snapshot depth (up to 32 snapshots per object)